REMARKS

In response to the Final Office Action dated October 19, 2007, Applicants respectfully request reconsideration based on the above claim amendments and the following remarks. Applicants respectfully submit that the claims as presented are in condition for allowance. Prior to entry of this response, Claims 1-18 and 20 were pending in the application, of which Claims 1, 8, and 14 are independent. In the Final Office Action dated October 19, 2007, Claims 1-18 and 20 were rejected under 35 U.S.C. § 103(a) and under 35 U.S.C. § 112. Following this response, Claims 1-6, 8-12, 14-16, 18, and 20 remain in this application with Claims 7, 13, and 17 being canceled without prejudice or disclaimer. Applicants hereby address the Examiner's rejections in turn.

I. Rejection of the Claims Under 35 U.S.C. § 112, Second Paragraph

In the Final Office Action dated October 19, 2007, the Examiner rejected Claims 1-18 and 20 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the Applicants regard as their invention. Claims 1, 8, and 14 have been amended to remove “a form name corresponding to each of the one or more field names of the form.” Applicants respectfully submit that the amendments to Claims 1, 8, and 14 overcome this rejection. Applicants respectfully request withdrawal of this rejection.

II. Rejection of the Claims Under 35 U.S.C. § 103(a)

In the Office Action, the Examiner rejected Claim 1-18 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,999,948 ("*Nelson*") in view of U.S. Patent No. 6,496,843 ("*Getchius*"). Claims 1, 8, and 14 have been amended, and Applicants respectfully submit that the amendments overcome this rejection and add no new matter.

Amended Claim 1 is patentably distinguishable over the cited art for at least the reason that it recites, for example, "maintaining an output table for storing the response data, wherein maintaining the output table comprises: receiving a request to publish the response data associated with each of the field names, and storing the response data associated with each of the field names in the output table, the output table having output table fields with names identical to the field names, wherein storing the response data in the output table comprises writing the response data to output table fields matching the field names of the form such that the response data is written to output table fields with names identically matching the field names." Amended Claims 8 and 14 each includes a similar recitation. Support for these amendments can be found in the specification at least page 2, lines 25-29 and page 12, lines 12-22.

Consistent with embodiments of the invention, an output table for storing form response data may have the same name as a corresponding form. (*See* specification, page 2, lines 27-28.) The output table may also have fields that have names identical to form field contained in the form. (*See* specification, page 2, lines 28-29.) The field names associated with the response data may be extracted from a submission. (*See* specification, page 12, lines 16-17.) The response data may be written to the output table fields matching the form page's field names. (*See* specification, page 12, lines 17-19.) In this manner, the response data may be written to the output table's fields corresponding to the identically named form fields. (*See* specification, page 12, lines 19-20.)

In contrast, *Nelson* at least does not disclose the aforementioned recitation from Claim 1. For example, *Nelson* merely discloses that a dynamic forms tool creates a widget by finding a DATAITEM with the same name as the widget. (*See* col. 8, lines 57-59.) In *Nelson*, a data item's attributes automatically determine a widget type to create. (*See* col. 8, lines 60-61.) Since the widget is an integer with enumeration stings and a user is allowed to change the value, dynamic forms automatically create a drop-list combo box widget. (*See* col. 8, lines 61-64.) The combo box's drop list portion is then populated with the enumeration choices and resized to display these choices without using scrollbars. (*See* col. 8, lines 64-67 and FIG. 4.) Therefore, *Nelson* merely discloses using an integer enumerated stings to create and resize a drop-list without using scrollbars. However, nowhere in *Nelson* does it disclose an output table having an output table field with names identical to field names such that response data is written to output table fields with names identically matching the field names. Rather *Nelson* is silent regarding storing the response data in an output table.

In addition, *Getchius* does not overcome *Nelson's* deficiencies. *Getchius* merely discloses that an external process may copy blob data from multiple tables in which an associated field name differs with each table. (*See* col. 52, lines 31-33.) The external process in *Getchius* uses data included in a temporary table 1242 to fetch or access the blob data associated with a particular table name and field name to subsequently index into each particular table name using an identifier to extract the actual blob data. (*See* col. 52, lines 38-42.) Therefore, *Getchius*, fetches blob data associated with a particular table and field name. However, nowhere in *Getchius* does it disclose an output table having an output table field with names identical to the field names such that response data is written to output table fields with names identically matching the field names. Rather *Getchius* is silent regarding storing the response data in an output table.

Combining *Nelson* with *Getchius* would not have led to the claimed invention because *Nelson* and *Getchius*, either individually or in any reasonable combination, at least do not disclose “maintaining an output table for storing the response data, wherein maintaining the output table comprises: receiving a request to publish the response data associated with each of the field names, and storing the response data associated with each of the field names in the output table, the output table having output table fields with names identical to the field names, wherein storing the response data in the output table comprises writing the response data to output table fields matching the field names of the form such that the response data is written to output table fields with names identically matching the field names,” as recited by amended Claim 1. Amended Claims 8 and 14 each includes a similar recitation. Accordingly, independent Claims 1, 8, and 14 each patentably distinguishes the present invention over the cited art, and Applicants respectfully request withdrawal of this rejection of Claims 1, 8, and 14.

Dependent Claims 2-6, 9-12, 15-16, and 18 are also allowable at least for the reasons described above regarding independent Claims 1, 8, and 14, and by virtue of their respective dependencies upon independent Claims 1, 8, and 14. Accordingly, Applicants respectfully request withdrawal of this rejection of dependent Claims 2-6, 9-12, 15-16, and 18.

III. Rejection of the Claims Under 35 U.S.C. § 103(a)

In the Office Action, the Examiner rejected Claim 20 under 35 U.S.C. § 103(a) as being unpatentable over *Nelson* in view of *Getchius* further in view of U.S. Patent No. 6,718515 ("*Conner*"). Dependent Claim 20 is patentably distinguishable over the cited art for at least the reason that it includes, due to its dependency on amended independent Claim 1, "maintaining an output table for storing the response data, wherein maintaining the output table comprises: receiving a request to publish the response data associated with each of the field names, and storing the response data associated with each of the field names in the output table, the output table having output table fields with names identical to the field names, wherein storing the response data in the output table comprises writing the response data to output table fields matching the field names of the form such that the response data is written to output table fields with names identically matching the field names."

As stated above, consistent with embodiments of the invention, an output table for storing form response data may have the same name as a corresponding form. (*See* specification, page 2, lines 27-28.) The output table may also have fields that have names identical to form field contained in the form. (*See* specification, page 2, lines 28-29.) The field names associated with the response data may be extracted from a submission. (*See* specification, page 12, lines 16-17.) The response data may be written to the output table fields matching the form page's field names. (*See* specification, page 12, lines 17-19.) In this manner, the response data may be written to the output table's fields corresponding to the identically named form fields. (*See* specification, page 12, lines 19-20.)

In contrast, and as stated above, *Nelson* at least does not disclose the aforementioned recitation from Claim 1. For example, *Nelson* merely discloses that a dynamic forms tool creates a widget by finding a DATAITEM with the same name as the widget. (*See* col. 8, lines 57-59.) In *Nelson*, a data item's attributes automatically determine a widget type to create. (*See* col. 8, lines 60-61.) Since the widget is an integer with enumeration stings and a user is allowed to change the value, dynamic forms automatically create a drop-list combo box widget. (*See* col. 8, lines 61-64.) The combo box's drop list portion is then populated with the enumeration choices and resized to display these choices without using scrollbars. (*See* col. 8, lines 64-67 and FIG. 4.) Therefore, *Nelson* merely discloses using an integer enumerated stings to create and resize a drop-list without using scrollbars. However, nowhere in *Nelson* does it disclose an output table having an output table field with names identical to field names such that response data is written to output table fields with names identically matching the field names. Rather *Nelson* is silent regarding storing the response data in an output table.

In addition, *Getchius* does not overcome *Nelson's* deficiencies. *Getchius* merely discloses that an external process may copy blob data from multiple tables in which an associated field name differs with each table. (*See* col. 52, lines 31-33.) The external process in *Getchius* uses data included in a temporary table 1242 to fetch or access the blob data associated with a particular table name and field name to subsequently index into each particular table name using an identifier to extract the actual blob data. (*See* col. 52, lines 38-42.) Therefore, *Getchius*, fetches blob data associated with a particular table and field name. However, nowhere in *Getchius* does it disclose an output table having an output table field with names identical to the field names such that response data is written to output table fields with names identically matching the field names. Rather *Getchius* is silent regarding storing the response data in an output table.

Furthermore, *Conner* does not overcome *Nelson's* and *Getchius'* deficiencies. *Conner* merely discloses a method for creating a table format object and using the object to generate an HTML table as a dynamic page in response to a client browser. (*See* col. 5, lines 11-14.) A routine in *Conner* begins by creating a table format object called a tableFormatter. (*See* col. 5, lines 14-16.) The object is created during a page authoring process. (*See* col. 5, lines 16-17.) In response to a client request, the request object and data object are passed, in *Conner*, to the tableFormatter that formats the table for use in a page. (*See* col. 5, lines 38-44.) In other words, in response to a client request in *Conner*, a .jsp servlet creates the HTML table. Then the servlet populates the table according to properties set in the tableFormatter that is hard-coded by a page author. *Conner* populates a table according to properties that are hard-coded and not retrieved. Like *Nelson* and *Getchius*, *Conner* at least does not disclose an output table having an output table field with names identical to the field names such that response data is written to output table fields with names identically matching the field names.

Combining *Nelson* with *Getchius* and *Conner* would not have led to the claimed invention because *Nelson*, *Getchius*, and *Conner*, either individually or in any reasonable combination, at least does not disclose “maintaining an output table for storing the response data, wherein maintaining the output table comprises: receiving a request to publish the response data associated with each of the field names, and storing the response data associated with each of the field names in the output table, the output table having output table fields with names identical to the field names, wherein storing the response data in the output table comprises writing the response data to output table fields matching the field names of the form such that the response data is written to output table fields with names identically matching the field names,” as included in dependent Claim 20. Accordingly, dependent Claim 20 patentably distinguishes the present invention over the cited art, and Applicants respectfully request withdrawal of this rejection of dependent Claim 20.

IV. Conclusion

In view of the foregoing remarks, Applicants respectfully request the reconsideration and reexamination of this application and the timely allowance of the pending claims. The preceding arguments are based only on the arguments in the Office Action, and therefore do not address patentable aspects of the invention that were not addressed by the Examiner in the Office Action. The claims may include other elements that are not shown, taught, or suggested by the cited art. Accordingly, the preceding argument in favor of patentability is advanced without prejudice to other bases of patentability. Furthermore, the Office Action contains a number of statements reflecting characterizations of the related art and the claims. Regardless of whether any such statement is identified herein, Applicant declines to automatically subscribe to any statement or characterization in the Office Action.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 13-2725.

Respectfully submitted,
MERCHANT & GOULD P.C.

P.O. Box 2903
Minneapolis, MN 55402-0903
404.954.5066

Date: October 31, 2007

/D. Kent Stier/

D. Kent Stier
Reg. No. 50,640

DKS:ARL:mdc

39262
PATENT TRADEMARK OFFICE